2 copy!

TABLE NO

MULTIMEDIA UNIVERSITY

STUDENT ID NO

					1	2	5	7	3

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2017/2018

TCP1201 OBJECT ORIENTED PROGRAMMING AND DATA STRUCTURES / TCP1241 COMPUTER PROGRAMMING II

(All sections/groups)

3 MARCH 2018
9:00 a.m. – 11:00 a.m.
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This question paper consists of 10 pages with 4 questions only.
2. Answer **ALL QUESTIONS**.
3. All questions carry equal marks (20 marks) and the distribution of the marks for each sub-question is given.
4. Write your answers in the question paper itself.

Question 1

a) Draw a UML class diagram showing:

- Cat is a subclass of Animal.
- Every animal has a name, and has an accessor and a mutator for it.
- Every animal has a method called *sound*, which takes a float value, *decibel*, for how loud to make the sound.
- Cat, in addition, has another method called *sound*, which takes *decibel* and a boolean called *hiss*. (The idea is that if this boolean is true, then the cat will hiss instead of making a normal meow.)
- Make sure that the attributes and methods have sensible visibilities and types.
- The constructor for Animal should be able to take a name. [4 marks]

b) Write the accessors and mutators, and the constructor for Cat. [4 marks]

Continued...

c) Would it be better for Animal to be a normal class, an abstract class, or an interface? Explain your reason. [3 marks]

d) Is the sound with the hiss in Cat a case of overriding or overloading? Explain the reason for your choice. [3 marks]

e) Here is the Java code for another Animal:

```
public class Snake extends Animal {
    public void sound(float decibel) {
        SoundPlayer.playSound("snakehiss.wav", decibel);
    }
}
```

Assume that there is also code provided for Cat, Snake, Dog, Eagle, Raccoon and Hamster.

Write a method, *soundAll*, that will receive a Collection of Animals, and *decibel* and use polymorphism to get all the all the animals to make their sound, one at a time.

[6 marks]

Continued...

Question 2

a) For the given code, explain if the code compiles and runs error-free all the time. If there is a need for any corrections, rewrite the correct version. [7 marks]

```
import java.util.*;
public class ExcepTest {

public static void main(String args[]) {
  Scanner scan = new Scanner(System.in);
  int a[] = {34, 67, 45};
  System.out.println(
     "Enter an index of the array you want to access:");
  System.out.println("The element is: " + a[scan.nextInt()]);

}
```

Hint: One problem is that Scanner's nextInt() throws InputMismatchException, and there may be other problems.

Continued...

b) Novice programmers might think only to use *if...else* statement to handle exceptions that may occur in a program. How would you convince the programmer to instead use the object-oriented exception handling mechanism? Provide one code sample of how a *throw* and *throws* can be used instead of an *if...else* statement. [7 marks]

Continued...

c) Write a recursive factorial program in Java. [6 marks]

Continued...

Question 3

a) Complete the following **MyStack** class which implements a Stack data structure with **ArrayList**, with the following methods:

push – Pushes an item into the stack.
pop – Pops the topmost item off the stack. Returns the value of the popped item.
getSize – Returns the size of the stack
copy – Copies the content of the stack to a new stack (passed in as input).

[10 marks]

Code:

```
import java.util.*;

public class MyStack<E> {
  private ArrayList<E> list = new ArrayList<>();

  // push method




  // pop method




  // getSize method




  // copy method









}
```

Continued...

b) Complete the program below which creates a **LinkedList** instance to store person names. The program first sorts the linked list (in ascending order), removes the second element from the list and prints it, then outputs the final list using the **ListIterator** interface. Use the **LinkedList** class provided by **java.util** library. The expected output of the program is given below.

[8 marks]

Expected output:

```
Jack was removed from list
Brian Siti Vinod
```

Code:

```
import java.util.*;
public class PersonList {
  public static void main(String[] args) {
    // create instance of LinkedList called linkedList



    // add four persons into the list
    linkedList.add("Vinod");
    linkedList.add("Brian");
    linkedList.add("Siti");
    linkedList.add("Jack");

    // sort the list


    // remove the second element in the list and print it out



    // outputs the list using ListIterator interface




}
```

Continued...

c) Describe an operation that is **more efficient** on an array-based list than a linked list.

[2 marks]

Question 4

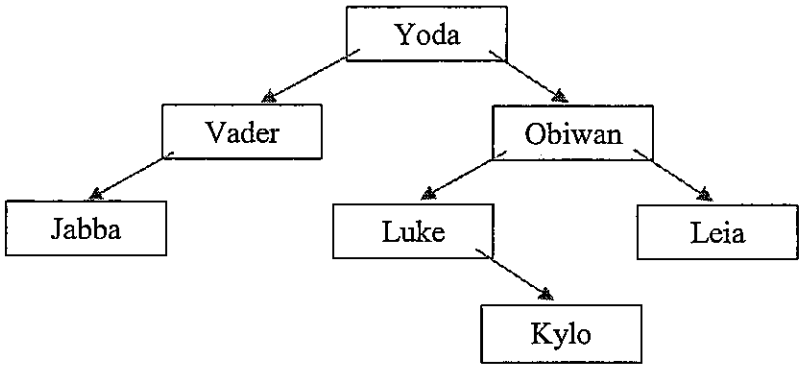
a) Write the pseudocode for inserting a node into a **Binary Search Tree (BST)**. You may leave out the implementation of the tree node.

[11 marks]

Continued...

b) In the BST given below, write the sequence of nodes visited based on **postorder traversal**. Assume the arrows pointing diagonally to the left and right directions as the left and right children, respectively.

[3 marks]



c) Describe what is a **Set** and explain its characteristics.

Write a few lines of code to first instantiate a **HashSet** object which takes in **String** content, add a string to the set, and finally print the contents of the set.

[6 marks]

End of paper